**Programs:**

**Merge Sort:**

#include<stdio.h>

#include<stdlib.h>

void mergeSort(int \*, int);

void merge(int\*, int, int\*, int);

void main(){

int \*arr, i, n;

printf("Enter the number of elements in the array :");

scanf("%d",&n);

arr = (int\*) malloc(sizeof(int));

printf("Enter the elements to be sorted: ");

for(i = 0;i < n; i++)

scanf("%d",&arr[i]);

mergeSort(arr, n);

printf("The sorted elements are : ");

for(i = 0;i < n; i++)

printf("%d\t",arr[i]);

printf("\n");

}

void mergeSort(int \*array, int size){

int mid;

if(size == 1)

return;

else{

mid = size/2;

mergeSort(array, mid);

mergeSort(array + mid, size - mid);

merge(array, mid, array + mid, size - mid);

}

}

void merge(int \*a, int s1, int \*b, int s2){

int i, j, k, \*temp\_arr;

temp\_arr = (int\*) malloc((s2+s1) \* sizeof(int));

i = j = k = 0;

while(i < s1 && j < s2)

temp\_arr[k++] = (a[i] < b[j]) ? a[i++] : b[j++];

while(i < s1)

temp\_arr[k++] = a[i++];

while(j < s2)

temp\_arr[k++] = b[j++];

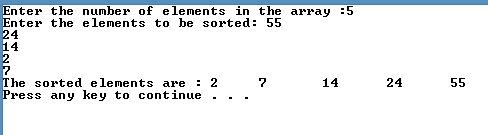
for(i = 0; i < k; i++)

a[i] = temp\_arr[i];

free(temp\_arr);

}

**Output:**



**Quick Sort:**

#include<stdio.h>

#include<conio.h>

void quickSort(int[], int, int);

void main() {

int arr[100], i, size;

printf("Enter the number of elements in the array :");

scanf("%d", &size);

printf("Enter the elements to be sorted: ");

for (i = 0;i < size; i++)

scanf("%d", &arr[i]);

quickSort(arr, 0, size - 1);

printf("The sorted elements are: ");

for (i = 0;i < size; i++)

printf("%d ", arr[i]);

\_getch();

}

void quickSort(int array[], int LB, int UB) {

int pivot, nxt\_pvt, left, right;

left = LB;

right = UB;

pivot = array[left];

while (LB < UB) {

while ((array[UB]) >= pivot && (LB < UB))

UB--;

if (LB != UB) {

array[LB] = array[UB];

LB++;

}

while ((array[LB]) <= pivot && (LB < UB))

LB++;

if (LB != UB) {

array[UB] = array[LB];

UB--;

}

}

array[LB] = pivot;

nxt\_pvt = LB;

LB = left;

UB = right;

if (LB < nxt\_pvt)

quickSort(array, LB, nxt\_pvt - 1);

if (UB > nxt\_pvt)

quickSort(array, nxt\_pvt + 1, UB);

}

**Output:**

